



Validation and Verification

Lecture Objectives:

- 1) Define validation and verification.
- 2) Explain the term IV & V.
- 3) Define the terms fault, error, and failure.
- 4) Define testing, test failure, and debugging.
- 5) Explain the requirements for a fault to be observed (Reachability, Infection, Propagation).
- 6) Explain the concept of test case values
- 7) Explain expected results
- 8) Explain test set.
- 9) Explain test script.

- What is verification?

Making sure it works.

Discussion

Initial Requirements
are met.

Make sure what was made is
what person wanted.

oh¹
✓

- What is validation?

Uhm
✓

Discussion

Validation and Verification

Definitions (IEEE)

- Validation

- The process of evaluating software at the end of software development to ensure compliance with intended usage

Needs Domain knowledge of the system.

- Verification

— "Testing Activities"

- The process of determining whether the products of a given phase of the software development process fulfill the requirements established during the previous phase

Technical skills.

Validation:

Are we building the
right product?

Verification

Are we building the
product right?

Independent Verification and Validation (IV&V)

- Independent *Someone else doing the work. ✓ V&V*
 - Activity is done by non-developers
- Verification
 - Testing of the software ✓
- Validation
 - Ensuring that the software meets the customer's needs

Government contracts.

Disclaimer...

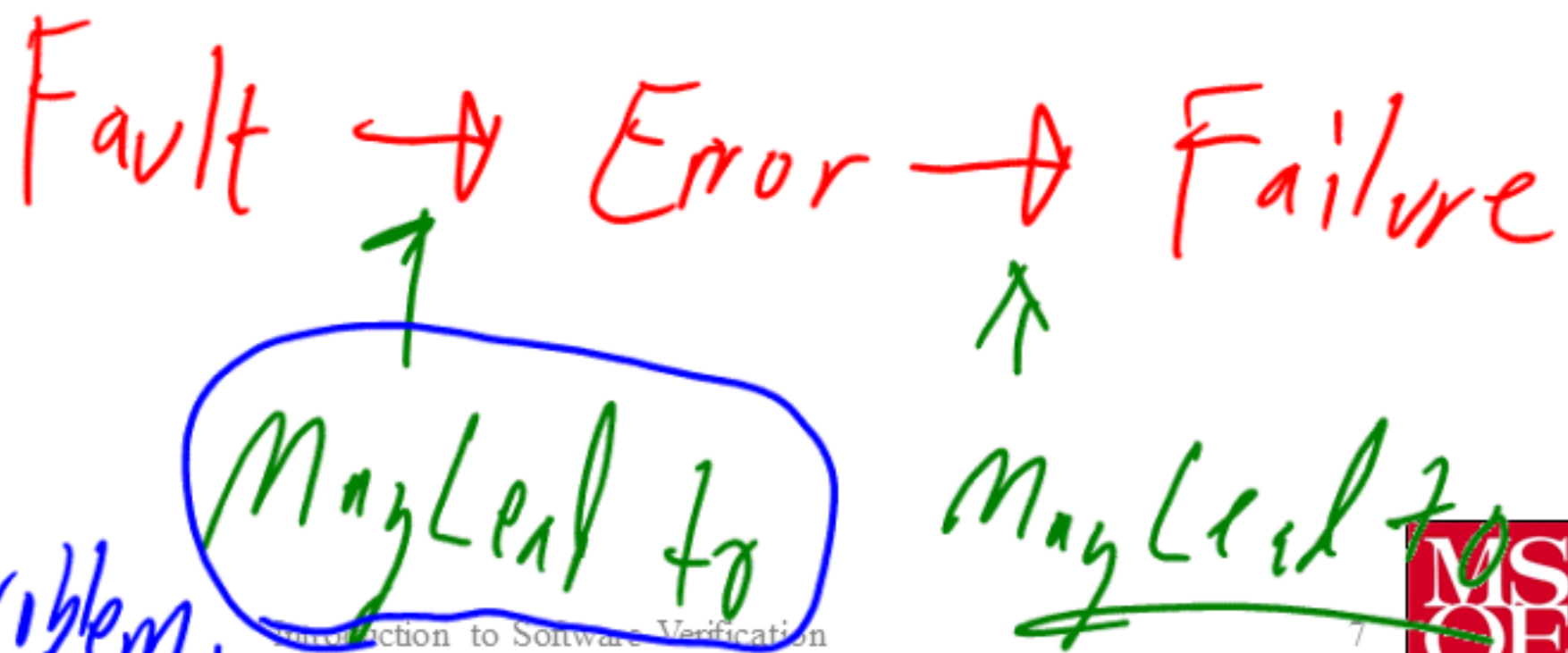
- Like any field, software testing comes with a large number of specialized terms that have particular meanings in this context
- Some of the following terms are standardized, some are used consistently throughout the literature and the industry, but some vary by author, topic, or test organization
- The definitions here are intended to be the most commonly used

"Mistake"
↳ instead of
↳ =

- Software Fault- A defect present in the software ⇒ wrong line of code
- Software Error – An incorrect internal state that is brought on by a software fault.
- Software Failure – an externally visible incorrect behavior of the software.

Definitions

Software
crash /
or other problem.



Discussion

- Why do faults cause failures?

Fault ~~prevents~~ causes something that shouldn't happen.

Invalid data state.

↳ null object

- Why do faults not cause failures?

Unused code =

Checks in code / exception handling

RIP Model

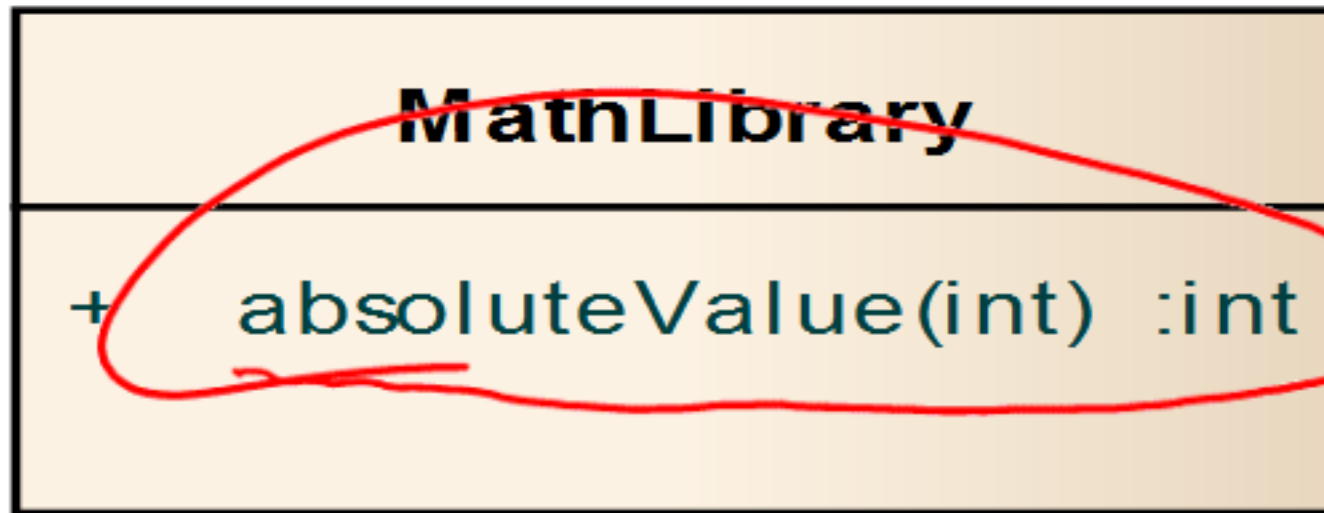
Why will a fault manifest itself as a failure?

1. Reachability: The location or locations in the program that contain the fault must be reached - *Not reached / executed \Rightarrow will not fail.*
2. Infection: The state of the program must be incorrect *Something must go wrong w/ the data.*
3. Propagation: The infected state must propagate to cause some output of the program to be incorrect

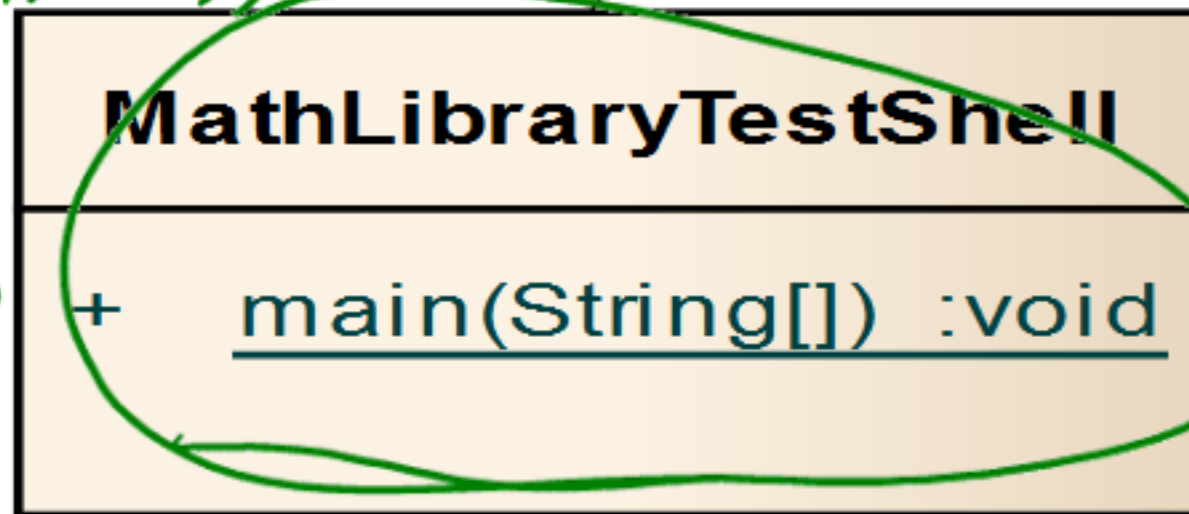
User behavior must be wrong.

Testing Demonstration

class Class Model



'test harness'



Text File w/ values.

Testing Definitions

- Test Case Values - *Inputs to program.*
 - The input necessary to complete some execution of the software under test
- Expected Results - *What it should see.*
 - The results which are to be produced if and only if the program satisfies the intended behavior
- Test Set
 - A set of test cases used to test software
- Executable Test Script - *Mechanism for running tests*
 - A test case that is prepared in a form to be executed automatically on the test software and produce a report.